

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions and listings of the claims in the present application.

Listing of Claims:

Claims 1-18 (Canceled).

19. (Previously Presented) A method for changing software in a first memory area in a control unit for controlling operational sequences, the method comprising:

replacing execution of old software parts with execution of new software parts; and

~~writing the old software parts into the first memory area;~~

overwriting at least a portion of the old software parts with an exit address that causes the performance of a first branching from the first memory area to a location in the new software parts whenever the overwritten portion is reached during execution of a corresponding old software part;

wherein the old software parts are written into the first memory area, the new software parts are written into a second memory area and, due to the first branching a first branching in the first memory area, instead of the old software parts being executed in the first memory area, the new software parts are executed in the second memory area, the control unit, following the execution of the new software parts, branching back again into the first memory area via a second branching in the second memory area and the execution of the other software distinct from the old software parts being continued in the first memory area, the old software parts remaining, at least in part, unmodified in the first memory area.

20. (Previously Presented) The method of claim 19, wherein the second memory area is only used to receive the new software parts.

21. (Previously Presented) The method of claim 19, wherein the first branching and the second branching are implemented by at least one chained list.

22. (Previously Presented) The method of claim 19, wherein as a first branching a start address of the new software parts is used, this being used to overwrite at least partially the old software parts.

23. (Previously Presented) The method of claim 19, wherein as the second branching a start address of the additional software distinct form the old software parts is used.

24. (Previously Presented) The method of claim 19, wherein the new software parts contain information that indicate which old software parts are to be replaced.

25. (Previously Presented) The method of claim 19, wherein the new software parts contain information that indicate by which new software parts the old software parts are to be replaced.

26. (Previously Presented) The method of claim 19, wherein the second memory area, in addition to at least one new software part, contains an address for the first branching, an address for the second branching and an address for the start of the old software part, which is to be replaced by the at least one new software part.

27. (Previously Presented) The method of claim 26, wherein the second memory area furthermore contains the length of the at least one new software part and of the at least one old software part.

28. (Previously Presented) The method of claim 27, wherein the addresses are integrated into a data record in the second memory area.

29. (Currently Amended) The method of claim 28, wherein at least two old software parts and [[the]] at least two new software parts, which replace these, are provided, and the data record is created and stored in the second memory area for each instance of an old software part having a corresponding new software part the addresses being respectively integrated into one data record and written into the second memory area.

30. (Previously Presented) The method of claim 19, wherein as the first memory area a first table and as the second memory area a second table are provided in the same memory.

31. (Currently Amended) The method of claim 19, wherein the first memory area and the second memory area are each divided into [[two]] software sections, a size of the software

section of the first memory area being equal to a size of the software section of the second memory area of equal size, a new software part being written into each software section of the second memory area.

32. (Previously Presented) The method of claim 28, wherein every data record or every software section is provided with an identification.

33. (Previously Presented) The method of claim 32, wherein the identification for a software section in the first memory area, which contains an old software part, and the identification for the corresponding software section having the new software part, which replaces the old software part, are the same.

34. (Currently Amended) A device for changing software in a first memory area in a control unit for controlling operational sequences, the device comprising:

~~a control unit for replacing the execution of old software parts with the execution of new software parts, and the old software parts being written into the first memory area, wherein an arrangement is included, which writes the new software parts into a second memory area and a first branching into the first memory area, whereby, instead of the old software parts being executed in the first memory area, the new software parts are executed in the second memory area, the arrangement also writing a second branching into the second memory area, whereby, following the execution of the new software parts, the control unit branches back again into the first memory area and the execution of the additional software distinct from the old software parts is continued in the first memory area, the old software parts remaining in the first memory area performing the following:~~

overwriting at least a portion of the old software parts with an exit address that causes the performance of a first branching from the first memory area to a location in the new software parts whenever the overwritten portion is reached during execution of a corresponding old software part;

wherein the old software parts are written into the first memory area, the new software parts are written into a second memory area and, due to the first branching, instead of the old software parts being executed in the first memory area, the new software parts are executed in the second memory area, the control unit, following the execution of the new software parts, branching back again into the first memory area via a second branching in the

second memory area and the execution of the other software distinct from the old software parts being continued in the first memory area, the old software parts remaining, at least in part, unmodified in the first memory area.

35. (Currently Amended) A control unit comprising:

a first memory area, in which old software parts and additional software parts distinct from the old software parts are stored;

a second memory area, which contains new software parts replacing the old software parts; and

an arrangement performing the following: to write the new software parts into a second memory area and a first branching into the first memory area, whereby, instead of the old software parts being executed in the first memory area, the new software parts are executed in the second memory area, the arrangement also writing a second branching into the second memory area, whereby, following the execution of the new software parts, the control unit branches back again into the first memory area and the execution of the additional software distinct from the old software parts is continued in the first memory area, the old software parts remaining in the first memory area

overwriting at least a portion of the old software parts with an exit address that causes the performance of a first branching from the first memory area to a location in the new software parts whenever the overwritten portion is reached during execution of a corresponding old software part;

wherein the old software parts are written into the first memory area, the new software parts are written into a second memory area and, due to the first branching, instead of the old software parts being executed in the first memory area, the new software parts are executed in the second memory area, the control unit, following the execution of the new software parts, branching back again into the first memory area via a second branching in the second memory area and the execution of the other software distinct from the old software parts being continued in the first memory area, the old software parts remaining, at least in part, unmodified in the first memory area.

36. (Currently Amended) A computer-readable storage medium storing a computer program executable in a control unit of a computer, the computer program comprising:

program code for changing software in a first memory area in a control unit for controlling operational sequences, by performing the following at the control unit:

replacing execution of old software parts with execution of new software parts; and

~~writing the old software parts into the first memory area;~~

overwriting at least a portion of the old software parts with an exit address that causes the performance of a first branching from the first memory area to a location in the new software parts whenever the overwritten portion is reached during execution of a corresponding old software part;

wherein the old software parts are written into the first memory area, the new software parts are written into a second memory area and, due to the first branching a first branching in the first memory area, instead of the old software parts being executed in the first memory area, the new software parts are executed in the second memory area, the control unit, following the execution of the new software parts, branching back again into the first memory area via a second branching in the second memory area and the execution of the other software distinct from the old software parts being continued in the first memory area, the old software parts remaining, at least in part, unmodified in the first memory area.

37. (New) The method of claim 28, further comprising:

restoring full execution of the old software parts by deleting the data record.